

CLAIMS

What is claimed is:

1. A method for creating and using a state machine, comprising:  
creating a first state;  
creating a second state;  
building a state machine from the first and second states, the state machine being capable of performing a plurality of functions; and  
using common code for each of the plurality of functions, the common code not being unique for a function of a given type.
2. The method of Claim 1, wherein the given type includes at least one of editing, storing, loading, and varying a parameter.
3. The method of Claim 1, wherein the plurality of functions include editing, storing, loading, and varying a parameter.
4. The method of Claim 1, further comprising recognizing dead states in the state machine.
5. The method of Claim 4, further comprising removing dead states from the state machine.
6. The method of Claim 4, further comprising reviving a dead state in the state machine.
7. The method of Claim 1, further comprising estimating a number of states needed to build the state machine.

8. The method of Claim 7, further comprising estimating a number of dead states in the state machine at a point in time after the state machine is built.
9. The method of Claim 1, further comprising, after building the state machine, running the state machine to test a device under test.
10. The method of Claim 1, further comprising, after building the state machine, running the state machine to test software under test.
11. The method of Claim 1, further comprising, after building the state machine, running the state machine to simulate a device.

12. A computer readable medium having a program of instructions implemented in code, the program of instructions comprising:
  - creating a first state;
  - creating a second state;
  - building a state machine from the first and second states, the state machine being capable of executing at least one function, the at least one function being implemented in code common to multiple parameters.
13. The computer readable medium of Claim 12, wherein the at least one function includes at least one of the group consisting of editing, storing, loading, and displaying.
14. The computer readable medium of Claim 12, wherein the multiple parameters include a Peripheral Component Interconnect (PCI) cache line.
15. The computer readable medium of Claim 12, wherein the multiple parameters include a Small Computer System Interface (SCSI) synchronous rate.
16. The computer readable medium of Claim 12, wherein the multiple parameters include a block size.
17. The computer readable medium of Claim 12, further comprising a look up table for storing default values of the multiple parameters.
18. The computer readable medium of Claim 12, further comprising a look up table for providing type and value information for each of the multiple parameters.
19. The computer readable medium of Claim 12, wherein the type and value information includes a range of values that are permitted for each of the multiple

parameters.

20. The computer readable medium of Claim 19, wherein the type and value information includes an incremental step size for each of the multiple parameters.

21. The computer readable medium of Claim 12, wherein at least one of the multiple parameters is independent of type.

22. A system that behaves in accordance with a state machine, comprising:  
a processor for implementing and using a state machine;  
a memory coupled to the processor, the memory storing information about the state machine;  
a bus; and  
a device coupled to the processor via the bus, the device capable of being tested by the state machine, wherein the state machine corresponds to a function that is stored in generic parameter independent code, wherein the state machine has a current state.
23. The system of Claim 22, wherein the state machine is implemented as a linked list of states.
24. The system of Claim 23, wherein each of the states corresponds to a state of the state machine.
25. The system of Claim 24, wherein the state machine receives inputs.
26. The system of Claim 25, wherein the inputs and the current state of the state machine determine a next state and an output of the state machine.
27. The system of Claim 26, wherein, for a given input, each state has a next state.
28. The system of Claim 27, wherein, for the given input, each state has a next next state.
29. The system of Claim 28, wherein, for the given input, each state has a previous state and a previous previous state.

30. A method for varying parameters for a device under test, comprising:  
setting a current parameter to a first parameter;  
setting a flag to indicate that testing is to continue; and  
determining if the current parameter is to be randomized.
31. The method of Claim 30, further comprising, if it is determined that the current parameter is to be randomized, then randomizing the current parameter's value index.
32. The method of Claim 31, further comprising, determining if the current parameter is equal to the last parameter.
33. The method of Claim 32, further comprising, if it is determined that the current parameter is not the last parameter, then setting the current parameter to the next parameter.
34. The method of Claim 33, further comprising, if the flag indicates that testing is to continue, determining if the current parameter's value is equal to the last parameter.
35. The method of Claim 34, further comprising, if it is determined that the current parameter is not to be randomized, then determining if the current parameter's value is equal to the last parameter.
36. The method of Claim 35, further comprising, if it is determined that the current parameter is the last parameter, then checking the flag to indicate that testing is to continue.
37. The method of Claim 36, further comprising, if checking the flag indicates that testing is to continue, testing the device under test with current parameter values.

38. The method of Claim 36, further comprising, if checking the flag indicates that testing is not to continue, then stopping testing of the device under test.

39. The method of Claim 37, further comprising, setting the current parameter to a new parameter for a new test cycle.

40. The method of Claim 32, further comprising, if it is determined that the current parameter is the last parameter, then setting the current parameter to a next parameter.

41. The method of Claim 40, further comprising determining if the current parameter is to be randomized.

42. The method of Claim 41, wherein a function that uses the current parameter is implemented in common code.

43. The method of Claim 42, wherein the current parameter is independent of type.